

Лабораторная работа 1.2

Xenomai Real Time Linux — Периодический режим запуска задачи.

Цель работы: Знакомство с Xenomai real time Linux, запуск простых задач в периодическом режиме.

Аппаратное и программное обеспечение: PC, ОС Linux, Virtual Box, Xenomai virtual box image.

Порядок выполнения работы:

1. Работа выполняется на локальной или удаленной виртуальной машине с установленным Xenomai. В случае выполнения работы на локальной виртуальной машине запустите VirtualBox и активируйте виртуальную машину с установленным Xenomai. Для входа в систему используйте имя пользователя **root** и пустой пароль. В случае использования удаленной виртуальной машины запустите терминал и введите команду:

```
ssh -l root ip_адрес_удаленной_машины
```

2. Если необходимо, создайте рабочий каталог, в качестве имени каталога необходимо указать фамилии выполняющих работу, например:

```
mkdir Ivanov_Petrova
```

3. В рабочем каталоге необходимо создать файл lab12.c содержащий код программы запуска простой задачи в периодическом режиме - **periodic mode** реального времени. Пример кода:

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <sys/mman.h>

#include <native/task.h>
#include <native/timer.h>

RT_TASK demo_task;

/* NOTE: error handling omitted. */

void demo(void *arg)
{
    RTIME now, previous;

    /*
     * Arguments: &task (NULL=self),
     *            start time,
     *            period (here: 1 s)
     */
    rt_task_set_periodic(NULL, TM_NOW, 500000000);
    previous = rt_timer_read();

    while (1) {
        rt_task_wait_period(NULL);
        now = rt_timer_read();

        /*
         * NOTE: printf may have unexpected impact on the timing of
         *       your program. It is used here in the critical loop
         *       only for demonstration purposes.
         */
    }
}
```

```

        */
        printf("Time since last turn: %ld.%06ld ms\n",
              (long)(now - previous) / 1000000,
              (long)(now - previous) % 1000000);
        previous = now;
    }
}

void catch_signal(int sig)
{
}

int main(int argc, char* argv[])
{
    signal(SIGTERM, catch_signal);
    signal(SIGINT, catch_signal);

    /* Avoids memory swapping for this program */
    mlockall(MCL_CURRENT|MCL_FUTURE);

    /*
     * Arguments: &task,
     *            name,
     *            stack size (0=default),
     *            priority,
     *            mode (FPU, start suspended, ...)
     */
    rt_task_create(&demo_task, "trivial", 0, 99, 0);

    /*
     * Arguments: &task,
     *            task function,
     *            function argument
     */
    rt_task_start(&demo_task, &demo, NULL);

    pause();

    rt_task_delete(&demo_task);
}

```

Выполните компилирование программы. Запустите программу (подробные инструкции по компилированию и запуску программ см в методических указаниях к лабораторной работе 1.1).

4. Добавьте в программу еще одну задачу с периодом запуска две секунды. Задача должна выводить в консоль фамилии выполняющих работу. Выполните компилирование и запустите программу.
5. Измените код программы таким образом, чтобы задача выводящая фамилии имела более высокий приоритет. Выполните компилирование и запустите программу.

Источники информации:

www.xenomai.org

www.google.ru